

## DEVICE AND METHOD FOR RETRIEVING DATA AND RECORDING MEDIUM

Patent Number: JP2000311088  
Publication date: 2000-11-07  
Inventor(s): NAKAJIMA YOICHI  
Applicant(s):: NEC CORP  
Requested Patent: ☐ JP2000311088 (JP00311088)  
Application Number: JP19990123058 19990428  
Priority Number(s):  
IPC Classification: G06F9/45  
EC Classification:  
Equivalents:

---

### Abstract

---

**PROBLEM TO BE SOLVED:** To replace data which coincides with desired data in terms of meaning with another kind of data by retrieving it.

**SOLUTION:** A data analyzing part 22 performs a syntactic analysis of contents in a data defining part 11 in accordance with the language specifications of a source program 1 and registers the analysis result in a data definition dictionary 21. A character string to be a replacement object and the character string after replacement are given as a parameter 3. A data collating part 23 performs the syntactic analysis of the character string to the replacement object which is given as the parameter 3 in accordance with the language specifications of the source program 1 and collates it with the syntactic analysis result which is registered in the dictionary 21. The data which is coincident in terms of meaning is extracted as the character string to be the replacement object as the result of collation and its position is reported to a character string replacement part 25. A character string replacement part 25 replaces the character string to be the replacement object, which is reported from the data collating part 23, with the character string after replacement, which is given as the parameter.

---

Data supplied from the **esp@cenet** database - I2

[illegible]

## 【特許請求の範囲】

【請求項 1】 所定のルールに従って記述された第 1 の文字列を、前記所定のルールに基づいて構文解析する第 1 の解析手段と、

検索対象となる第 2 の文字列を、前記所定のルールに基づいて構文解析する第 2 の解析手段と、

前記第 1 の解析手段による解析結果中に、前記第 2 の解析手段による解析結果と意味的に一致する部分があるかどうか照合する第 1 の照合手段と、

前記第 1 の照合手段による照合で意味的に一致した部分に対応する前記第 1 の文字列中の文字列を検索された文字列として抽出する第 1 の抽出手段とを備えることを特徴とするデータ検索装置。

【請求項 2】 前記第 2 の文字列が置換されるべき第 3 の文字列を指示する第 1 の指示手段と、

前記第 1 の抽出手段によって抽出された文字列を、前記第 1 の指示手段から指示された第 3 の文字列に置換する第 1 の置換手段とをさらに備えることを特徴とする請求項 1 に記載のデータ検索装置。

【請求項 3】 前記第 1 の抽出手段は、前記第 1 の照合手段による照合で意味的に一致した部分に対応する前記第 1 の文字列中の文字列の位置を前記第 1 の置換手段に通知し、

前記第 1 の置換手段は、前記第 1 の抽出手段から通知された文字列の位置に基づいて、前記第 3 の文字列への置換を行うことを特徴とする請求項 2 に記載のデータ検索装置。

【請求項 4】 前記第 1 の解析手段は、前記第 1 の文字列の解析結果に、前記指示手段から指示された前記第 3 の文字列への置換を行うための情報を対応付けさせ、前記第 1 の置換手段は、前記第 1 の文字列の解析結果に対応付けられている情報に基づいて、前記第 3 の文字列への置換を行うことを特徴とする請求項 2 または 3 に記載のデータ検索装置。

【請求項 5】 前記第 1、第 2 の解析手段は、それぞれ前記第 1、第 2 の文字列を語に分割し、分割した各語を所定のデータ構造で関係付けてそれぞれの解析結果とし、前記第 1 の照合手段は、前記第 1、第 2 の解析手段による解析結果を、それぞれに含まれる語を単位として比較しながら、意味的に一致する部分があるかどうか照合することを特徴とする請求項 1 乃至 4 のいずれか 1 項に記載のデータ検索装置。

【請求項 6】 前記第 1、第 2 の文字列は、階層構造で記述された複数の語を含み、

前記第 1、第 2 の解析手段は、階層構造で記述された複数の語を、上位側または下位側の語から順次リストでつなげてそれぞれの解析結果とし、

前記第 1 の照合手段は、前記第 1、第 2 の解析手段による解析結果を、それぞれのリストをたどっていきながら、意味的に一致する部分があるかどうか照合すること

を特徴とする請求項 5 に記載のデータ検索装置。

【請求項 7】 前記第 1 の解析手段は、階層構造の最上位または最下位の語を、ハッシュ法によって得られたハッシュ値に従った所定の記憶位置に記憶させ、

前記第 1 の照合手段は、前記第 2 の文字列中の階層構造で記述された語のうちの最上位または最下位の語からハッシュ値を求め、当該ハッシュ値の位置に記憶されている第 1 の文字列中の語から順にリストをたどっていきながら、意味的に一致する部分があるかどうかを照合することを特徴とする請求項 6 に記載のデータ検索装置。

【請求項 8】 前記第 1 の文字列は、所定のプログラミング言語で記述されており、

前記第 1、第 2 の解析手段は、前記所定のプログラミング言語の仕様に基づいて、前記第 1、第 2 の文字列をそれぞれ構文解析することを特徴とする請求項 1 乃至 7 のいずれか 1 項に記載のデータ検索装置。

【請求項 9】 前記第 1 の文字列は、前記所定のプログラミング言語におけるデータ定義であり、前記所定のプログラミング言語における処理記述である第 4 の文字列を、オペランドに分割するオペランド解析手段と、

前記オペランド解析手段によって分割されたオペランドを構文解析する第 3 の解析手段と、

前記第 3 の解析手段によるオペランドの解析結果に、前記第 1 の解析手段の解析結果と意味的に一致する部分があるかどうか照合する第 2 の照合手段と、

前記第 1 の照合手段による照合で意味的に一致した部分に対応する前記第 4 の文字列中の文字列を検索された文字列として抽出する第 2 の抽出手段とをさらに備えることを特徴とする請求項 8 に記載のデータ検索装置。

【請求項 10】 前記第 3 の解析手段は、前記オペランド解析手段によって分割されたオペランドのうちで、予約語以外のオペランドを構文解析の対象とすることを特徴とする請求項 9 に記載のデータ検索装置。

【請求項 11】 前記第 4 の文字列が置換されるべき第 5 の文字列を指示する第 2 の指示手段と、

前記第 2 の抽出手段によって抽出された文字列を、前記第 2 の指示手段から指示された第 5 の文字列に置換する第 2 の置換手段とをさらに備えることを特徴とする請求項 9 または 10 に記載のデータ検索装置。

【請求項 12】 所定のルールに従って記述された第 1 の文字列を、前記所定のルールに基づいて構文解析する第 1 の解析ステップと、

検索対象となる第 2 の文字列を、前記所定のルールに基づいて構文解析する第 2 の解析ステップと、

前記第 1 の解析ステップでの解析結果中に、前記第 2 の解析ステップでの解析結果と意味的に一致する部分があるかどうか照合する照合ステップと、

前記照合ステップにおける照合で意味的に一致した部分に対応する前記第 1 の文字列中の文字列を、検索された

文字列として抽出する抽出ステップとを含むことを特徴とするデータ検索方法。

【請求項 13】前記第 2 の文字列が置換されるべき第 3 の文字列を指示する指示ステップと、前記抽出ステップで抽出された文字列を、前記第 1 の指示ステップで指示された第 3 の文字列に置換する置換ステップとをさらに含むことを特徴とする請求項 12 に記載のデータ検索方法。

【請求項 14】所定のルールに従って記述された第 1 の文字列を、前記所定のルールに基づいて構文解析する第 1 の解析ステップと、検索対象となる第 2 の文字列を、前記所定のルールに基づいて構文解析する第 2 の解析ステップと、前記第 1 の解析ステップでの解析結果中に、前記第 2 の解析ステップでの解析結果と意味的に一致する部分があるかどうか照合する照合ステップと、前記照合ステップにおける照合で意味的に一致した部分に対応する前記第 1 の文字列中の文字列を、検索された文字列として抽出する抽出ステップとを実行させるためのプログラムを記録することを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 15】前記第 2 の文字列が置換されるべき第 3 の文字列を指示する指示ステップと、前記抽出ステップで抽出された文字列を、前記第 1 の指示ステップで指示された第 3 の文字列に置換する置換ステップとをさらに実行させるためのプログラムを記録することを特徴とする請求項 14 に記載のコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、意味的に一致するかどうかの照合によって、データを検索するデータ検索装置、方法及び記録媒体に関する。

【0002】

【従来の技術】従来の文字列置換システムは、ソースプログラム等のテキストの内容を取得して、指定した文字列と一致する文字列があるかどうかの照合処理を、テキストの先頭から順に行い、一致する文字列があった場合には、その文字列を指定した文字列と置換するという処理を、テキストの末尾まで繰り返し行っている。

【0003】しかしながら、従来のシステムでは、指定した文字列と置換対象の文字列との照合処理が、「語」の概念を意識せずに単なる文字列同士の比較として行っているため、本来は置換したくない文字列でも置換対象となってしまうという問題がある。例えば、“かき”を“くり”に置換しようとした場合には、“かきくけこ”という文字列中の“かき”も置換対象となってしまう。

【0004】そこで、特開平 3-150677 号公報において、プログラムを単語（語、字句、トークン）に分割し、分割された単語を単位として検索文字列との比較

を行う文字列検索方式が提案されている。この技術を採用すれば、“かき”を“くり”に置換しようとしたときに、“かきくけこ”という文字列中の“かき”が置換対象とされることはなくなる。

【0005】

【発明が解決しようとする課題】しかしながら、上記公報に記載の技術においても、指定した文字列と意味的には一致していても、表現方法が一致しない文字列は、置換対象とされないという問題がある。例えば、“う”の上位に“た”が存在する階層関係を“う OF た”で表すものとし、“う”の上位に“な”が存在し、“な”の上位に“た”が存在する階層関係を“う OF な OF た”で表すものとする。

【0006】このとき、“う OF た”と“う OF な OF た”は、意味的には同じになり、置換対象としてもよいはずである。ところが、従来のシステム（上記公報に記載のものを含む）では、これら両者の表現に一致する部分がないため、置換対象の文字列として取り扱うことができない。このように、本来は置換対象としたい文字列でも、置換対象とならないという問題がある。

【0007】本発明は、上記従来技術の問題点を解消するためになされたものであり、他のデータへの置換対象となるデータなど、所望のデータと意味的に一致するデータを検索することができるデータ検索装置、方法及びこの方法を実行するためのプログラムを記録した記録媒体を提供することを目的とする。

【0008】

【課題を解決するための手段】上記目的を達成するため、本発明の第 1 の観点にかかるデータ検索装置は、所定のルールに従って記述された第 1 の文字列を、前記所定のルールに基づいて構文解析する第 1 の解析手段と、検索対象となる第 2 の文字列を、前記所定のルールに基づいて構文解析する第 2 の解析手段と、前記第 1 の解析手段による解析結果中に、前記第 2 の解析手段による解析結果と意味的に一致する部分があるかどうか照合する第 1 の照合手段と、前記第 1 の照合手段による照合で意味的に一致した部分に対応する前記第 1 の文字列中の文字列を検索された文字列として抽出する第 1 の抽出手段とを備えることを特徴とする。

【0009】上記データ検索装置では、第 1 の照合手段は、単なる文字列の表現としてではなく、構文解析した結果に基づいて第 1 の文字列と第 2 の文字列とに一致があるかどうかを照合している。これにより、文字列の表現として異なっている、同一の意味を有する文字列を検索された文字列として第 1 の抽出手段が抽出することが可能となる。

【0010】上記データ検索装置は、前記第 2 の文字列が置換されるべき第 3 の文字列を指示する第 1 の指示手段と、前記第 1 の抽出手段によって抽出された文字列

を、前記第1の指示手段から指示された第3の文字列に置換する第1の置換手段とをさらに備えるものとしてもよい。

【0011】ここで、前記第1の抽出手段は、前記第1の照合手段による照合で意味的に一致した部分に対応する前記第1の文字列中の文字列の位置を前記第1の置換手段に通知してもよい。この場合、前記第1の置換手段は、前記第1の抽出手段から通知された文字列の位置に基づいて、前記第3の文字列への置換を行うものとすることができる。

【0012】さらに、前記第1の解析手段は、前記第1の文字列の解析結果に、前記指示手段から指示された前記第3の文字列への置換を行うための情報を対応付けさせるものとしてもよい。この場合、前記第1の置換手段は、前記第1の文字列の解析結果に対応付けられている情報に基づいて、前記第3の文字列への置換を行うものとすることができる。

【0013】上記データ検索装置において、前記第1、第2の解析手段は、それぞれ前記第1、第2の文字列を語に分割し、分割した各語を所定のデータ構造で関係付けてそれぞれの解析結果とすることができる。この場合、前記第1の照合手段は、前記第1、第2の解析手段による解析結果を、それぞれに含まれる語を単位として比較しながら、意味的に一致する部分があるかどうか照合することができる。

【0014】ここで、前記第1、第2の文字列は、階層構造で記述された複数の語を含むものであってもよい。この場合、前記第1、第2の解析手段は、階層構造で記述された複数の語を、上位側または下位側の語から順次リストでつなげてそれぞれの解析結果とし、前記第1の照合手段は、前記第1、第2の解析手段による解析結果を、それぞれのリストをたどっていきながら、意味的に一致する部分があるかどうか照合するものとすることができる。

【0015】さらに、前記第1の解析手段は、階層構造の最上位または最下位の語を、ハッシュ法によって得られたハッシュ値に従った所定の記憶位置に記憶させるものとし、前記第1の照合手段は、前記第2の文字列中の階層構造で記述された語のうちの最上位または最下位の語からハッシュ値を求め、当該ハッシュ値の位置に記憶されている第1の文字列中の語から順にリストをたどっていきながら、意味的に一致する部分があるかどうかを照合するものとするすることができる。

【0016】なお、上記データ検索装置において、前記第1の文字列は、所定のプログラミング言語で記述されたものとすることができる。この場合において、前記第1、第2の解析手段は、前記所定のプログラミング言語の仕様に基いて、前記第1、第2の文字列をそれぞれ構文解析するものとすることができる。

【0017】また、前記第1の文字列は、前記所定のプ

ログラミング言語におけるデータ定義である場合には、上記データ検索装置は、前記所定のプログラミング言語における処理記述である第4の文字列を、オペランドに分割するオペランド解析手段と、前記オペランド解析手段によって分割されたオペランドを構文解析する第3の解析手段と、前記第3の解析手段によるオペランドの解析結果に、前記第1の解析手段の解析結果と意味的に一致する部分があるかどうか照合する第2の照合手段と、前記第1の照合手段による照合で意味的に一致した部分に対応する前記第4の文字列中の文字列を検索された文字列として抽出する第2の抽出手段とをさらに備えるものとすることができる。

【0018】ここで、前記第3の解析手段は、前記オペランド解析手段によって分割されたオペランドのうちで、予約語以外のオペランドを構文解析の対象とすることができる。

【0019】このように第4の文字列をさらに解析するものにあつては、前記第4の文字列が置換されるべき第5の文字列を指示する第2の指示手段と、前記第2の抽出手段によって抽出された文字列を、前記第2の指示手段から指示された第5の文字列に置換する第2の置換手段とをさらに備えてもよい。

【0020】上記目的を達成するため、本発明の第2の観点にかかるデータ検索方法は、所定のルールに従って記述された第1の文字列を、前記所定のルールに基づいて構文解析する第1の解析ステップと、検索対象となる第2の文字列を、前記所定のルールに基づいて構文解析する第2の解析ステップと、前記第1の解析ステップでの解析結果中に、前記第2の解析ステップでの解析結果と意味的に一致する部分があるかどうか照合する照合ステップと、前記照合ステップにおける照合で意味的に一致した部分に対応する前記第1の文字列中の文字列を、検索された文字列として抽出する抽出ステップとを含むことを特徴とする。

【0021】上記データ検索方法は、前記第2の文字列が置換されるべき第3の文字列を指示する指示ステップと、前記抽出ステップで抽出された文字列を、前記第1の指示ステップで指示された第3の文字列に置換する置換ステップとをさらに含むものとすることができる。

【0022】上記目的を達成するため、本発明の第3の観点にかかるコンピュータ読み取り可能な記録媒体は、所定のルールに従って記述された第1の文字列を、前記所定のルールに基づいて構文解析する第1の解析ステップと、検索対象となる第2の文字列を、前記所定のルールに基づいて構文解析する第2の解析ステップと、前記第1の解析ステップでの解析結果中に、前記第2の解析ステップでの解析結果と意味的に一致する部分があるかどうか照合する照合ステップと、前記照合ステップにおける照合で意味的に一致した部分に対応する前記第1の文字列中の文字列を、検索された文字列として抽出する

抽出ステップとを実行させるためのプログラムを記録することを特徴とする。

【0023】上記コンピュータ読み取り可能な記録媒体は、前記第2の文字列が置換されるべき第3の文字列を指示する指示ステップと、前記抽出ステップで抽出された文字列を、前記第1の指示ステップで指示された第3の文字列に置換する置換ステップとをさらに実行させるためのプログラムを記録するものとするができる。

【0024】

【発明の実施の形態】以下、添付図面を参照して、本発明の実施の形態について説明する。

【0025】〔第1の実施の形態〕図1は、この実施の形態に係るデータ置換システムを示す機能ブロック図である。同図に示すように、このデータ置換システムでは、置換される文字列を含むソースプログラム1と、置換される文字列とその置換後の文字列を含むパラメータ3とが与えられる。また、システムとして、文字列変換機能2を含み、これが与えられたパラメータ3に従って、ソースプログラム1中の文字列を置換する。

【0026】ソースプログラム1は、所定のプログラミング言語（高級言語）に従って記述され、データ定義部11と、処理記述部12とを含む。パラメータ3は、置換される文字列（置換対象となる文字列）と置換後の文字列（置換されるべき文字列）とを指定する。

【0027】文字列置換機能2は、データ定義の辞書21と、データ解析部22と、データ照合部23と、オペランド解析部24と、文字列置換部25と、データ・処理照合部26との各機能を備えている。文字列置換機能2において、データ解析部22→データ照合部23→オペランド解析部24→文字列置換部25、の順に処理が行われるように制御されている。

【0028】データ定義の辞書21は、定義された個々のデータに対して一つずつの情報が登録される。データの階層構造に応じて、辞書上も階層構造を伴って登録されるように構築する。例えば、“あ”という名前のデータが複数回定義されていた場合でも、上位データが“か”であった場合と“さ”であった場合では、異なるデータであるということが辞書上で表現される。

【0029】データ解析部22は、ソースプログラム1中のデータ定義部11を構文解析し、データの定義内容を語（字句）間の関係の階層構造としてデータ定義の辞書21に登録する（これを、データ定義の辞書21の構築とする）。

【0030】データ照合部23は、データ定義の辞書21の構築後、パラメータ3として与えられた置換対象となる文字列（被置換文字列）を取得し、これを構文解析する。データ照合部23は、被置換文字列の構文解析結果をデータ定義の辞書21に登録されている構文解析結果と比較し、意味的に一致するデータの定義があるかどうかを照合する。データ照合部23は、上記照合の結果、被置換文字列と意味的に一致したデータの定義について、データ定義の辞書21上で、置換対象のデータであることを示す情報を設定する。

果、被置換文字列と意味的に一致したデータの定義について、データ定義の辞書21上で、置換対象のデータであることを示す情報を設定する。

【0031】オペランド解析部24は、被置換文字列照合後、ソースプログラム1中の処理記述部12を字句解析し、処理記述の内容を、構文的に独立した単位（オペランド）に分割する。

【0032】データ・処理照合部25は、オペランド解析部24が分割したオペランドを受けて、ソースプログラム1のプログラミング言語による予約語でないオペランドが、データ定義の辞書21中に存在するデータ名と一致するかの照合を行う。

【0033】文字列置換部26は、データ・処理照合部27が行った照合処理において照合に成功したオペランドについて、パラメータ3として与えられた置換後の文字列でソースプログラム1の処理記述部12を置換する。また、文字列置換部26は、データ照合部23が実施した照合において、パラメータ3として与えられた被置換文字列（置換対象の文字列）との照合に成功したデータについて、パラメータ3として与えられた置換後の文字列で置換する。すなわち、ソースプログラム1のデータ定義部11における照合に成功したデータは、置換後の文字列に置換される。

【0034】次に上述したシステムの機能を実現するハードウェアの構成について、図2のブロック図を参照して説明する。図示するように、このハードウェアは、二次記憶装置51と、メインメモリ52と、入力部53と、出力部54と、中央処理装置（以下、CPUという）55と、を備えており、これらの各構成要素は、バス56を介してそれぞれ接続されている。

【0035】二次記憶装置51は、例えばハードディスク等で構成されており、上記ソースプログラム1、及び上記文字列置換機能2を遂行させるためのプログラム、ファイル形式のデータ定義の辞書21を格納する。

【0036】メインメモリ52は、例えばRAM等から構成されており、二次記憶装置51から読み出されるソースプログラム1及び文字列置換機能2を遂行させるためのプログラムを記憶する記憶領域52A、52Bと、置換処理において必要なデータを一時的に記憶するワークエリア52Cと、が割り当られる。なお、上記データ定義の辞書21は、ワークエリア52C上で作成され、必要に応じて二次記憶装置51に保存される。また、パラメータ3、及びこの構文解析結果も、一時的に、ワークエリア52Cに記憶される。

【0037】入力部53は、キーボード、マウス等から構成されており、例えばパラメータ3を指定する。出力部54は、ディスプレイ装置から構成されており、例えばソースプログラム1、及び指定されたパラメータ3の表示、置換処理後のソースプログラム1の表示を行う。

【0038】CPU55は、バス55を介して上記各構

成要素を制御すると共に、二次記憶装置 5 1 からメインメモリ 5 2へプログラムをロードし、このプログラムを実行することにより、文字列置換機能 2 に含まれる各機能を実現する。

【0039】以下、この実施の形態にかかるデータ置換システムの各部の動作を、その処理順序に従って説明する。なお、ここでは、ソースプログラム 1 を記述するプログラミング言語が COBOL である場合を例にとりて説明する。

【0040】COBOL では、言語仕様上、“DATA DIVISION.” と記述される箇所と “PROCEDURE DIVISION.” と記述される箇所のとの間にデータが定義されるので、この間の記述をデータ定義部 1 1 とする。一方、“PROCEDURE DIVISION.” と記述される箇所の後ろに処理が記述されるので、この後ろの記述を処理記述部 1 2 とする。

【0041】COBOL におけるデータ定義及び処理記述を、“あ△OF△さ。”を例にとり、説明する。ここで、△は空白を意味する。この△は、説明の都合上、表記したものであり、実際には、COBOL による記述において使用されない。この記述における意味は、次の通りである。

【0042】(1) 「空白」は分離記号として扱われる。

(2) “OF” は階層関係を表す語を意味する。上記記述例では、“あ” の上位に “さ” が存在する階層関係が表現されている。

(3) ピリオド (.) は命令の終了を意味する。上記記述例では “あ OF さ” にてオペランドが確定されることになる。

【0043】次に、データ解析部 2 2 の処理について、図 3 を参照して説明する。データ解析部 2 2 は、COBOL の仕様に従って、ソースプログラム 1 中のデータ名が定義されている箇所であるデータ定義部 1 1 を処理対象とする。データ解析部 2 2 は、データ定義部 1 1 に文字列で記述することにより定義されたデータを、以後の照合処理に適用し易いよう、メモリ上のデータ定義の辞書 2 1 に登録する。

【0044】ここで、以後の照合処理に適用し易い登録の形式とは、(1) データの階層構造を辿れるように、従属するデータ及び従属されるデータの相互に参照が可能なポインタを持っていること、(2) データ名の検索が素早く行えるよう、ハッシュ法を用いてデータ名からハッシュ値を求め、そのハッシュ値が同じデータを連続して参照することが可能なポインタを持っていること、などの特徴を有する形式である。

【0045】例えば、図 3 中のデータ定義部 1 1 に記述されている文字列 (記述) を、データ解析部 2 2 が解析する場合を例に説明する。「01 か。//02 あ。

(//は改行を表す)」という文字列 (記述) を、デー

タ解析部 2 2 が、構文解析により、語 “あ” の上位に語 “か” が存在する階層関係であるという解析結果を得る。

【0046】そして、データ解析部 2 2 は、“あ” 及び “か” のデータそれぞれに、階層関係を表すポインタが付加すると共に、“あ” データに、ハッシュ法によりデータ名 (“あ”) から求めたハッシュ値に対応するポインタを付加する。

【0047】同様にして、2 番目、2 番目の文字列 (記述) についても解析が実施されると、図 3 中のデータ定義の辞書 2 1 に示すような階層構造の情報が得られる。

【0048】次に、データ照合部 2 3 の処理について、図 4 を参照して説明する。データ照合部 2 3 は、パラメータ 3 に与えられた被置換文字列 (置換対象の文字列) が、データ定義の辞書 2 1 中において、どのデータの定義に相当するかを決定する。

【0049】データ照合部 2 3 は、被置換文字列の内容を、ソースプログラム 1 の言語 (ここでは、COBOL) の仕様に従って構文解析し、解析結果として分割された先頭の語のハッシュ値を求め、データ定義の辞書 2 1 上で、そのハッシュ値で表される名標 (名前) を持つデータの定義を次々と走査し、先頭の語と一致するデータ名のデータを仮に確定する。その後、データ照合部 2 3 は、被置換文字列の記述内容に階層構造の表現がある場合は、この階層構造とデータ定義の辞書 2 1 上での階層構造とが一致しているか否かの照合処理を行う。

【0050】例えば、被置換文字列が “あ OF さ” となっている場合、“あ” の上位データとして “さ” が存在するかどうかは、データ定義の辞書 2 1 上で、階層構造を参照するポインタを “あ” を示すデータ名から上位に辿って行き、各層におけるデータ名と、“あ” を示すデータ名との照合処理により実施される。そして、“あ” を示すデータ名の上位に “さ” を示すデータ名が存在する場合には、そのデータ名は置換対象である、ということが確定される。

【0051】図 4 において、記号「×」は、パラメータ 3 からの “あ OF さ” の “あ” と、データ定義の辞書 2 1 上の “か” の下位に存在する “あ” とが最初に照合されたが、この階層構造で表現される文字列が、“あ OF さ” と意味的に一致しなかったことを示している。また、記号「○」は、パラメータ 3 からの “あ OF さ” の “あ” と、データ定義の辞書 2 1 上の “さ” の下位に存在する “あ” とが次に照合された結果、この階層構造で表現される文字列が、“あ OF さ” と意味的に一致したことを示している。なお、後述する図において表記される記号「×」及び「○」についても同様の意味とする。

【0052】データ照合部 2 3 は、上述したようにして置換対象であると確定されたデータ (データ名) に、データ定義の辞書 2 1 上で、置換対象であるということを

示す情報を設定する。

【0053】図4に示す例においては、確定されたデータ名（“あ”）に記号「◎」が付加されているが、この記号「◎」は、置換対象であるということを示す情報を意味している。なお後述する図において表記される記号「◎」についても同様の意味とする。

【0054】データ照合部23は、さらに、文字列置換部24に対して、意味的に一致したデータ定義部11中の文字列の開始位置と終了位置を通知して、データ定義部11中の該当するデータ定義の置換処理を指示する。

【0055】次に、オペランド解析部24の処理について、図5を参照して説明する。オペランド解析部24は、COBOL言語のに従って、ソースプログラム1中の処理記述が定義されている箇所である処理記述部12を処理対象とする。オペランド解析部24は、次のように処理を行う。

【0056】（1）言語仕様上の分離記号を基に、処理記述部12に文字列で記述された処理記述から「語」を分離して確定する。

（2）分離した「語」が言語仕様上の予約語ではない利用者語であった場合に、COBOL言語で階層関係を示す語“OF”を手掛かりに、処理記述中の利用者語の階層関係に従って、文字列の範囲を「オペランド」として確定する。

（3）確定した「オペランド」の単位で、置換対象であるか否かの判断処理を実施するよう、データ・処理照合部26に指示する。

【0057】ここで、処理記述中に“MOVE△100△TO△あ△OF△さ。”（△は空白文字とする）で示される文字列が記述されている場合において、“あ”以後の“あ△OF△さ。”の記述に対する解析について説明する。

【0058】オペランド解析部24は、最初に、分離記号を探す。COBOL言語では「空白」は分離記号として扱われるので、まず“あ”が「語」（字句）として確定される。後続に“OF”があるので、さらに後続の語“さ”で階層関係を表現していることが判断される。そして、“さ”の後ろにはピリオド（.）があり、これはCOBOLにおいては命令の終了を意味するので、“あ△OF△さ”の範囲でオペランドが確定されることになる。

【0059】このようにしてオペランドを確定したオペランド解析部24は、そのオペランド“あ△OF△さ”を、データ・処理照合部26に渡して、置換対象であるか否かの判断を実行させる。

【0060】次に、データ・処理照合部26の処理について、図6を参照して説明する。データ・処理照合部26は、オペランド解析部25から渡されたオペランドを語に分割し、データ照合部23がパラメータ3として与えられた被置換文字列と同様に構文解析する。次に、オペランドの先頭の語のハッシュ値を求め、データ定義の

辞書21上で、そのハッシュ値で表される名標を持つデータの定義を次々と走査し、先頭の語と一致するデータ名のデータを仮に確定する。

【0061】例えば、オペランドの最初の語が“あ”の場合は、“あ”と同じハッシュ値のデータを参照するポインタを次々に辿って行き、データ名との照合処理を行い、“あ”というデータ名を持つデータを仮に確定する。

【0062】その後、データ・処理照合部26は、オペランドの記述内容に階層構造の表現がある場合は、この階層構造とデータ定義の辞書21上での階層構造とが一致しているか否かの照合処理を行う。

【0063】例えば、オペランドが“あ OF さ”となっている場合、“あ”の上位データとして“さ”が存在するかどうかは、データ定義の辞書21上で、階層構造を参照するポインタを“あ”を示すデータ名から上位に辿って行き、各層におけるデータ名と、“あ”を示すデータ名との照合処理により実施される。そして、“あ”を示すデータ名の上位に“さ”を示すデータ名が存在する場合には、そのデータ名は置換対象である、ということが確定される。

【0064】このように、オペランドを示す文字列を、文字列全体で一度に照合せずに「語」毎にデータ定義の辞書21と照合することにより、例えば“あ△OF△さ”と“あ△△OF△さ”の二つの記述（△を空白文字とする）の様に、“あ”と“OF”の間の空白文字の数が記述箇所によって異なるような場合であっても、構文的なチェックの結果、同じデータ名を示す記述である、ということ判断することができる。

【0065】また、“う OF た”（2階層）と“う OF な OF た”（3階層）のように、階層構造の指定が異なるような場合であっても、意味的なチェックの結果、同じデータ名を示す記述である、ということ判断することができる。

【0066】置換対象であるということが確定されたデータに対して、データ照合部23による照合処理において置換対象であるという情報（図6の例では記号「◎」）が設定されている場合は、そのオペランドが置換対象（被置換文字列）となる。

【0067】この場合、データ・処理照合部26は、文字列置換部25に対して、当該オペランドのソースプログラム1中での開始位置と終了位置を指定し、処理記述12中の該当する処理記述の置換を指示する。

【0068】次に、文字列置換部25の処理について、図7を参照して説明する。文字列置換部25に対しては、データ照合部23、及びデータ・処理照合部26の両方から、文字列置換を実施するよう指示されるが、いずれの指示においても、置換される文字列のソースプログラム1中で開始位置と終了位置と置換文字列が通知される。



【0069】文字列置換部25は、この文字列置換の指示内容を、データ照合部23とデータ・処理照合部26の処理が終了するまで蓄積し、蓄積終了後、開始位置を降順に整列する。そして、文字列置換部25は、開始位置と終了位置で示されたソースプログラム1中の文字列を、パラメータ3として与えられた置換語の文字列で置換する。

【0070】なお、被置換文字列の開始位置の降順に整列するのは、次の様な理由からである。文字列を置換した際に、置換後の文字列の長さが置換前の文字列の長さ  
10と相違する場合には、置換後の文字列の後ろにある記述が相対的にずれる（後方にずれる）ことになる。そこで、ソースプログラム1の後ろから置換していくことによって、置換対象位置が順次前に移動することになるので、記述の相対的な移動を、後続の置換処理中に意識する必要がないからである。従って、処理記述12が先に置換処理され、その後、データ定義部11が置換処理されることになる。

【0071】なお、図7において、処理記述部12においては、「10行30桁～10行31桁」の“あ OF さ”が“ア OF さ”に置換されていることが示されており、またデータ定義部11においては、「5行15桁～5行16桁」の“あ”が“ア”に置換されていることが示されている。

【0072】以上説明したように、この実施の形態によれば、ソースプログラム1中の文字列が置換対象として適切であるということを、意味的に確認した上で決定するようにしているので、データ名単位での文字列置換を漏れなく正確に実施することができる。

【0073】また、上述した様に置換対象の文字列の絞り込みを正確に行うことができるので、文置換処理を起動した後に、誤った置換対象の文字列が置換された場合に、その文字列を元の文字列に戻すという、人手による作業を介在させる必要がなくなる。従って、複数のソースプログラム1に対する文字列置換を行う場合でも、オペレータは、最初にパラメータ3を指定するのみで良く、ごくわずかな作業量で一度に行うことができる。

【0074】〔第2の実施の形態〕図8は、第2の実施の形態に係る文字列検索装置の構成を示す機能ブロック図である。文字列置換部25が参照する置換語の文字列は、パラメータ3として直接渡されているが、この実施の形態では、置換語の文字列に関する情報を、一旦、データ定義の辞書21に保管し、文字列置換部25に対する置換指示の一つとして与える点で、第1の実施の形態と異なる。すなわち、データ照合部23及び文字列置換部25の処理が、第1の実施の形態での処理と多少異なっている。そこで、これらの構成要素について説明する。

【0075】パラメータ3に与えられた被置換文字列が、データ定義の辞書21中の、どのデータの定義に相

当するかを、データ照合部23によって決定する部分は、図4を参照して説明した処理と基本的に同様である。しかし、データ照合部23の処理は、次の2点について第1の実施の形態の場合と異なる。

【0076】図4に示す例では、確定されたデータに対するデータ定義の辞書21上での情報設定は、置換対象であるということを示す情報（図4では記号「◎」で示される情報）のみであったが、この実施の形態では、データ照合部23は、この情報に加えて、データ定義の辞書21に、図9に示すように、置換語の文字列が何であるかということを示す情報21Aを設定する。この情報21Aは、置換語の文字列が、例えば“ア OF さ”の場合は、“ア”を示す情報である。

【0077】また、データ照合部23から文字列置換部25への指定する内容は、被置換文字列（例えば“あ OF さ”）の開始位置と終了位置の情報のみであったが、この実施の形態では、データ照合部23は、これらの情報に加えて、置換文字列（例えば“ア OF さ”の“ア”）を指示内容として文字列置換部25に通知する。

20 【0078】文字列置換部25においては、図7を参照して説明した場合と異なる点は次の一点である。図7に示す例では、置換文字列はパラメータ3から直接取得するようにしているが、この実施の形態では、文字列置換部25には、図10に示すように、パラメータ3から一旦、データ定義の辞書21に格納された置換後の文字列に関する情報（例えば“ア OF さ”の場合は、“ア”）が、置換指示毎に与えられる。

30 【0079】この実施の形態によれば、置換対象となる文字列の情報と置換後の文字列に関する情報を、データ定義の辞書21中で一対にして記憶させているため、異なるパラメータ3の内容（置換前の文字列及び置換後の文字列）を、複数件同時に指定することができる。なお、この実施の形態においても、上記第1の実施の形態における効果と同様の効果を得ることができる。

40 【0080】〔第3の実施の形態〕上記の第1、第2の実施の形態では、階層構造で定義されたデータを置換する場合を例として説明した。これに対し、この実施の形態では、数式を置換する場合を例として説明する。ここでは、図11に示すような、(1)～(3)の数式を含むプログラムを例とする。

50 【0081】コンパイラなど、所定のプログラムを実行する処理装置が、数式(1)～(3)をそれぞれ読み込んだとき、これを構文解析すると、図12(a)～(c)に示すような構文木が得られる。数式(1)の構文木は、図12(a)に示すように、最上位の親のノードを「=」とし、左側の子のノードとして「a」を、右側の子のノードとして部分木101を有している。従って、数式(1)以降で、部分木101と意味的に同じ部分木が表れた場合には、その部分木を「a」に置き換えられることを意味しており、処理装置は、部分木101

と意味的に一致する構文木を「a」に置換するように指示する。

【0082】なお、部分木101は、親のノードを「+」とし、左側の子のノードを「b」、右側の子のノードを「c」としているが、親のノードが「+」であるので、左右の子のノードに序列はなく、子のノードとして「b」と「c」とを有していることとなる。

【0083】また、数式(2)の構文木は、図12(b)に示すように、部分木102を含んでいる。部分木102は部分木101と完全に一致するため、これを「a」に置換することができる。従って、数式(2)は、処理装置によって「 $d = e + a$ 」と置換される。

【0084】一方、数式(3)の構文木は、図12(c)に示すように、部分木103を含んでいる。部分木103は、親のノードが「+」であるため、左右の子のノードに序列はなく、子のノードとして「b」と「c」とを有していることとなる。従って、部分木103は部分木101と意味的に一致することとなり、数式(3)は、処理装置によって「 $f = g + a$ 」と置換される。

【0085】比較例として、従来の技術に従って同様の置換を行った場合には、数式(2)は、数式(1)に含まれる「 $b + c$ 」をその順番のまま含んでいるので、「 $d = e + a$ 」と置換することができる。しかし、数式(3)に含まれている「 $c + b$ 」は、数式(1)に含まれる「 $b + c$ 」と意味的に同じでも、文字列の表現として一致していないため、置換することができない。

【0086】従って、図11に示したプログラムに対して置換を行った後のプログラムを実行した場合の加算演算のステップ数は、この実施の形態では3ステップで済むのに対して、従来例では4ステップが必要となる。つまり、この実施の形態によれば、プログラムをより実行効率のよい形に最適化することができるようになり、プログラムの実行速度を向上させることが可能となる。

【0087】【実施の形態の変形】本発明は、上記の第1～第3の実施の形態に限られず、種々の変形、応用が可能である。以下、本発明に適用可能な上記の実施の形態の変形態様について、説明する。

【0088】上記の第1、第2の実施の形態では、COBOLで記述されたプログラム中に含まれる階層構造を定義した文字列を他の文字列に置換する場合を例として説明した。上記の第3の実施の形態では、所定のプログラミング言語で記述されたプログラム中に含まれる数式の一部に、演算結果が先に求められるものがある場合に、当該数式の一部を先の演算結果が代入される変数に置換する場合を例として説明した。しかしながら、本発明で適用可能なプログラミング言語は、COBOLに限られず、他の高級言語としてもよい。さらには、自然言語で記述された文章中の文字列と指定した文字列とをそれぞれ形態素解析し、形態素解析した結果同士を比較し

て、文章中の文字列を置換対象として判断してもよい。

【0089】上記の第1～第3の実施の形態では、指定した文字列から置換対象と判断された文字列を他の文字列に置換する場合を例として説明した。しかしながら、本発明は、文字列の置換を行わない場合であっても、指定した文字列と意味的に一致する文字列を検索するあらゆる場合に適用することができる。

【0090】上記の第1～第3の実施の形態で説明した処理を実行するためのプログラム(文字列置換機能2を実現するプログラム)は、CD-ROMなどのコンピュータ読み取り可能な記録媒体に格納して配布することも可能である。

【0091】

【発明の効果】以上説明したように、本発明によれば、文字列の表現として異なっても、同一の意味を有する文字列を、例えば他の文字列への置換対象の文字列などとして、検索することができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態に係るデータ置換装置システムの構成を示す機能ブロック図である。

【図2】図1に示したデータ置換システムを実現するハードウェアの構成を示すブロック図である。

【図3】データ解析部の処理を説明するための図である。

【図4】データ照合部の処理を説明するための図である。

【図5】オペランド解析部の処理を説明するための図である。

【図6】データ・処理照合部の処理を説明するための図である。

【図7】文字列置換部の処理を説明するための図である。

【図8】本発明の第2の実施の形態に係るデータ置換装置システムの構成を示す機能ブロック図である。

【図9】図8に示したデータ照合部の処理を説明するための図である。

【図10】図8に示した文字列置換部の処理を説明するための図である。

【図11】本発明の第3の実施の形態で適用されるプログラム中の数式例である。

【図12】(a)～(c)は、それぞれ図11の数式を構文解析した構文木を示す図である。

【符号の説明】

2 文字列置換機能

21 データ定義辞書

22 データ解析部

23 データ照合部

24 オペランド解析部

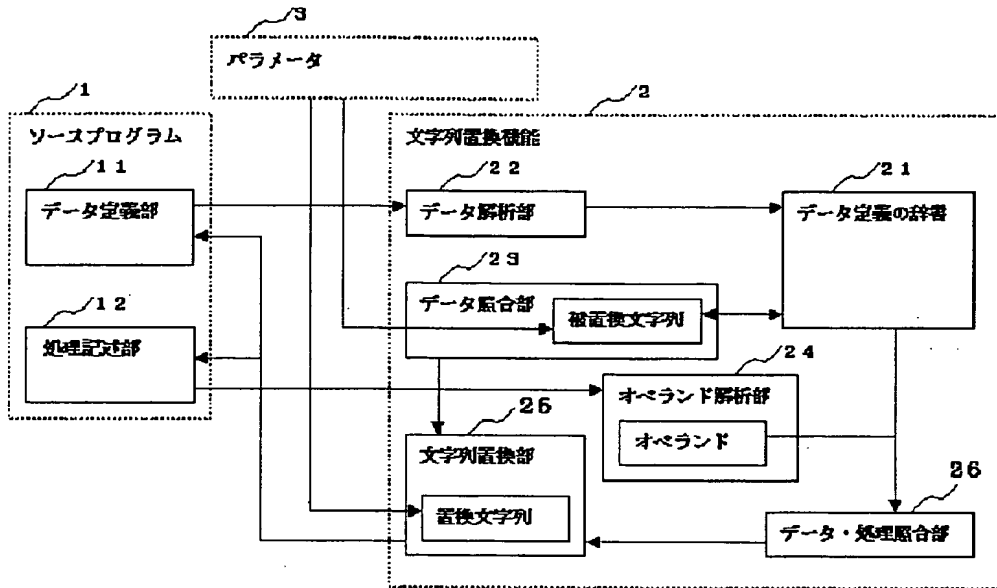
25 文字列置換部

50 26 データ・処理照合部

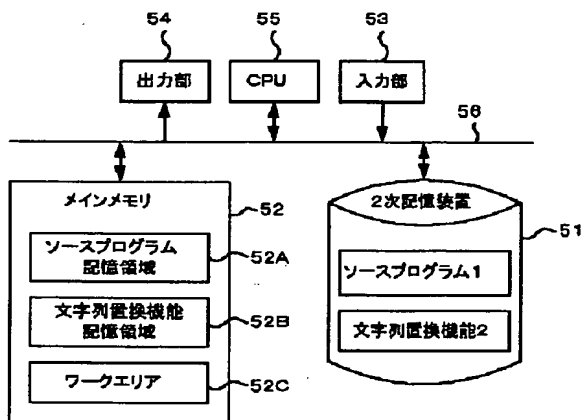
51 2次記憶装置  
52 メインメモリ  
53 入力部

54 出力部  
55 中央処理装置 (CPU)  
56 バス

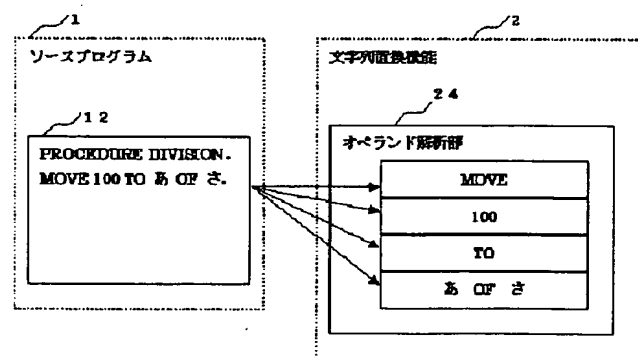
【図1】



【図2】



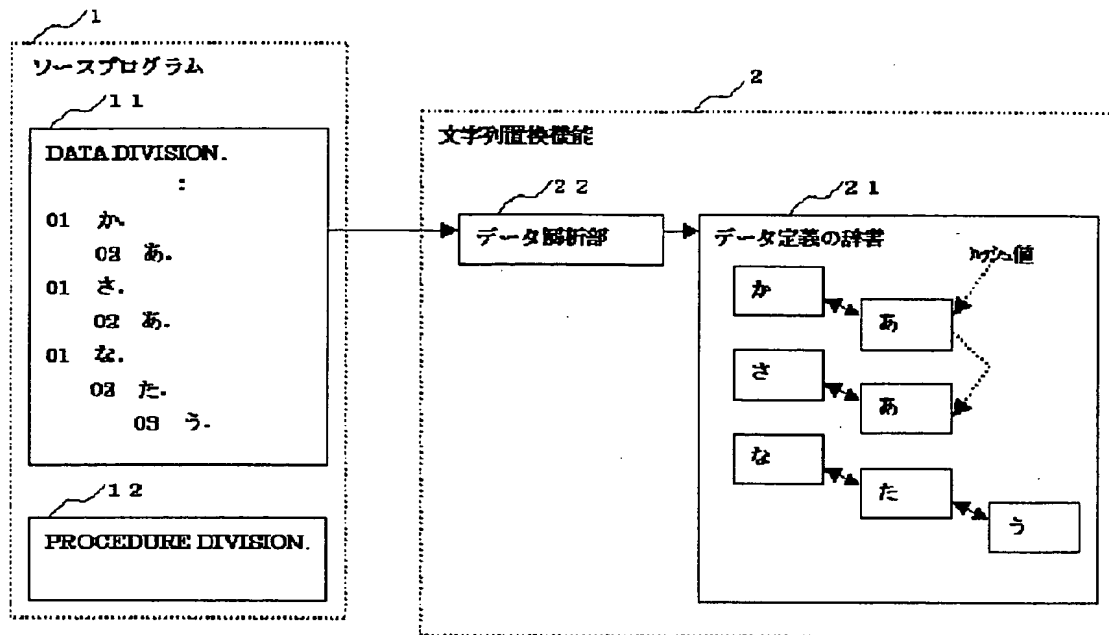
【図5】



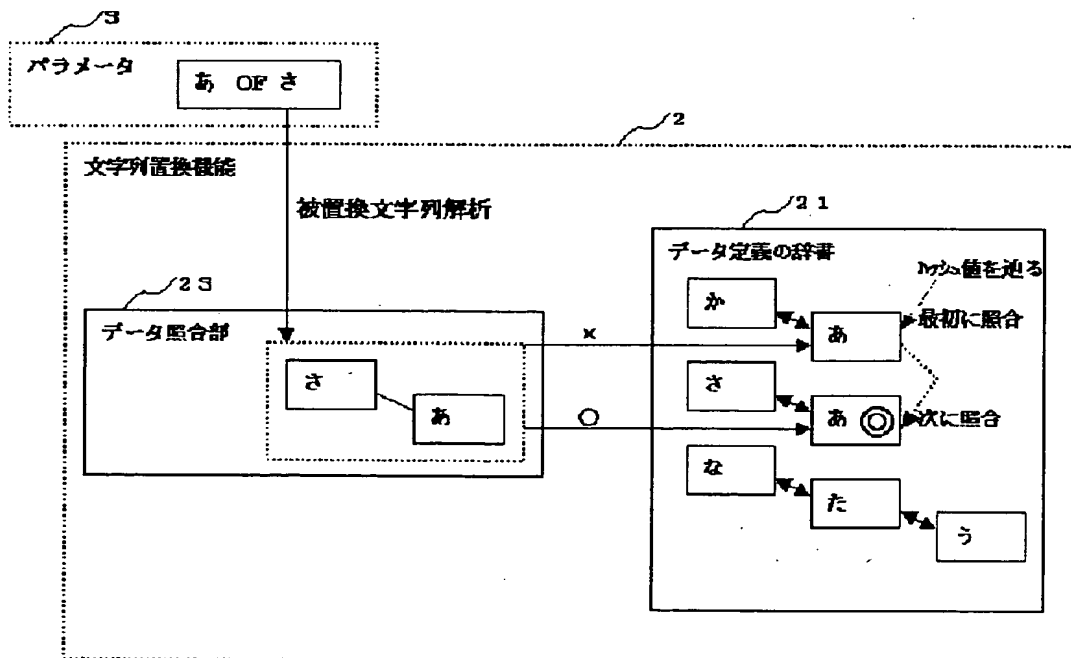
【図11】

$a=b+c$  ——— (1)  
 $d=e+b+c$  ——— (2)  
 $f=g+c+b$  ——— (3)

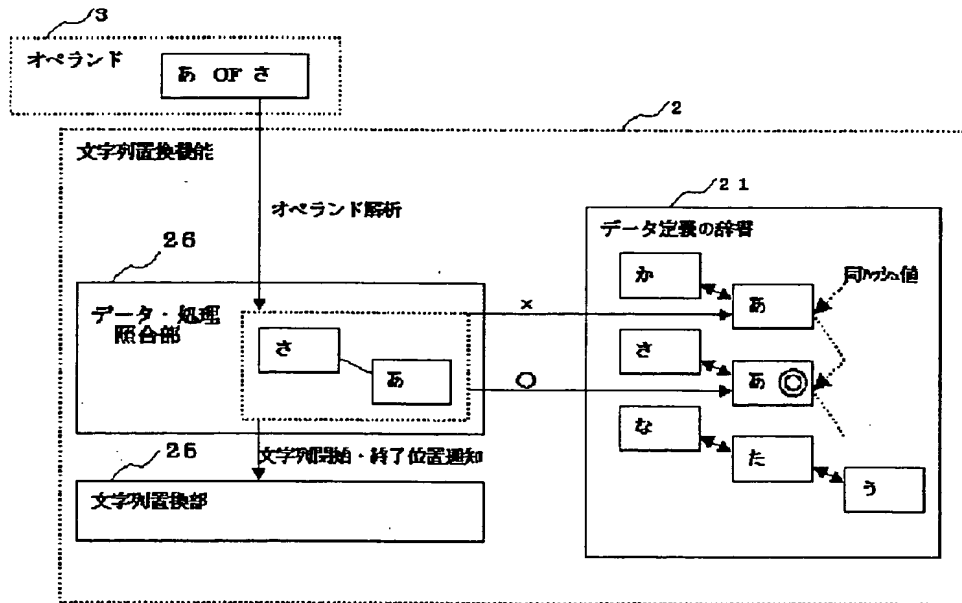
【図 3】



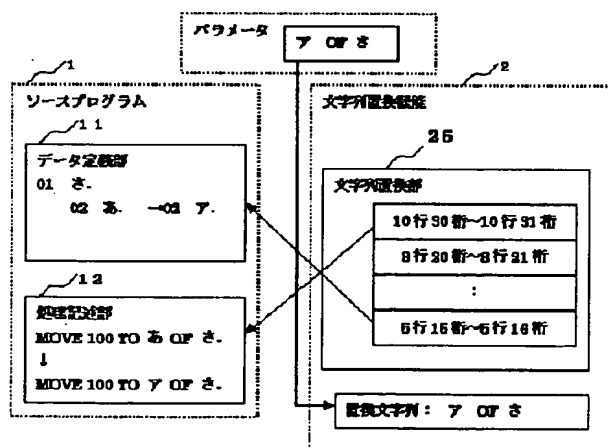
【図 4】



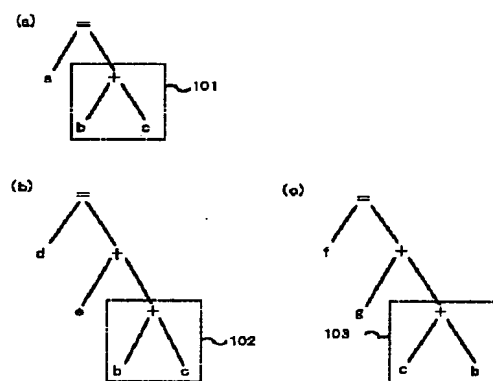
【図 6】



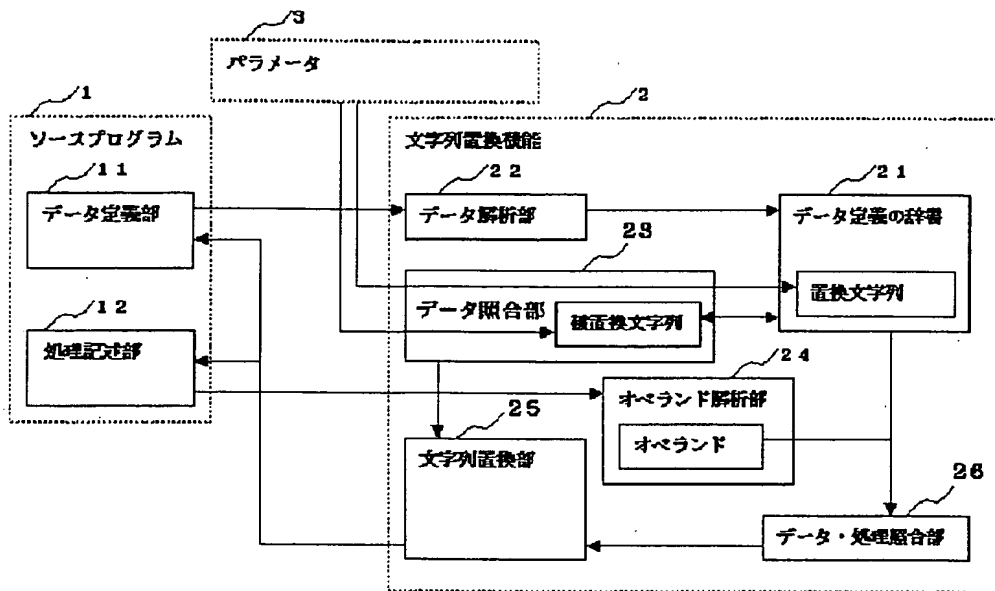
【図 7】



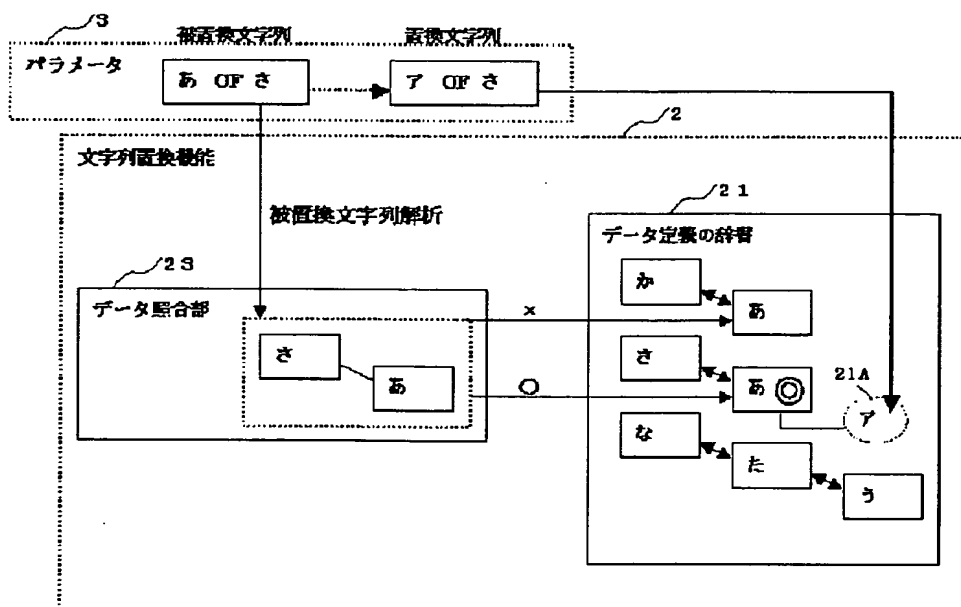
【図 12】



【図 8】



【図 9】



【図10】

